Recoder Builder Module Specification

James Francis

MIT 64-30111.0501

Rev. C

June 10, 1993

Contents

1 Introduction 5
1.1 Scope 5
1.2 References 5
1.3 Notations 5
1.4 Module Perspective 6
1.5 Document Overview 7
1.6 Acronyms 7

2 Overview 8
2.1 Purpose 8
2.2 Services 8
2.3 Module Context 9
2.4 Theory of Operation 9
2.5 Data Structures and Constants 10
2.6 Functions 10

3 Data Structures and Constants 11
3.1 Hardware 11
3.1.1 Recoder Table (0x048000 flat address) 11
3.1.2 PCA Recoder Input Event Word (14-bits) 11
3.1.3 Recoder Table Entry (16-bits) 12
3.1.4 Recoder Output Word (15 of 16 bits) 12
3.2 Recoder Parameter Set Structures 13
3.2.1 Pulse Height Correction Parameter Set (80 bytes) 13
3.2.2 Pulse Height Grouping Parameter Set (from 2 to 274 bytes) 15
3.2.3 Output Position Parameter Set (18 bytes) 15
3.3 Constants 16
3.3.1 RECODER_CHECK_DEFAULT (0x0ffff) 16
3.3.2 RECODER_NUM_RAW_ENERGIES (256) 16
3.3.3 RECODER_NUM_VIRTUAL (8) 16
3.3.4 RECODER_NUM_DETECTORS (5) 17
3.3.5 RECODER_NUM_DETECTOR_IDS (8) 17
3.3.6 RECODER_ADDR_PULSE_HEIGHT_MASK (0x000ff) 17
3.3.7 RECODER_ADDR_PULSE_HEIGHT_SHFT (0) 17
3.3.8 RECODER_ADDR_DETECTOR_ID_MASK (0x00700) 17
3.3.9 RECODER_ADDR_DETECTOR_ID_SHFT (8) 17
3.3.10 RECODER_ADDR_TYPER_SELECT_MASK (0x03800) 17
3.3.11 RECODER_ADDR_TYPER_SELECT_SHFT (11) 17
3.3.12 RECODER_CONT_OUTPUT_MASK (0x07fff) 17
3.3.13 RECODER_CONT_DISCARD_MASK (0x08000) 18

3.3.14 RECODER_POS_VIRT_MASK (0x0000f) 18
3.3.15 RECODER_POS_VIRT_SHFT (0) 18
3.3.16 RECODER_POS_CODE_MASK (0x000f0) 18
3.3.17 RECODER_POS_CODE_SHFT (4) 18
3.3.18 RECODER_POS_DET_MASK (0x00f00) 18
3.3.19 RECODER_POS_DET_SHFT (8) 18
3.3.20 RECODER_POS_OUTBAND_MASK (0x0f000) 18
3.3.21 RECODER_POS_OUTBAND_SHFT (12) 18
3.3.22 MSG_RECODER_DECODE_GROUPINGS_NULL 19
3.3.23 MSG_RECODER_DECODE_POSITION_NULL 19

4 Recoder Table Construction 20
4.1 Construction Algorithm 20
4.2 Pulse Height Correction 21
4.3 Pulse Height Grouping 22
4.4 Output Position 23

5 Functions 25
5.1 RECODERbuild () - Build Recoder Table Entries 25
5.2 RECODERencoded_lengths() - Lengths of Parameter Sets 26
5.3 RECODERcheck() - Check Integrity of Recoder Table 27

6 Detailed Design Considerations 28
6.1 Data Flow 28
6.2 Control Considerations 29
6.2.1 Pulse Height Correction Algorithm 29
6.2.2 Pulse Height Grouping Algorithm 29
6.3 Module Structure 30
6.3.1 Global Function Structures 30
6.3.2 Internal Function Structures 31
6.4 Optimization Considerations 31

7 Requirements Traceability 32
7.1 Science Requirements 32

8 Testing Considerations 33
8.1 Unit Testing 33
8.2 Integration Testing 34

Appendix A 36
A.1 Items Under Review 36

Figures

Figure 1 Module Perspective 6

Figure 2 Recoder Builder Module Context Diagram 9
Figure 3 Pulse Height Recoding 9
Figure 4 Recoder Table Input Event Word 11
Figure 5 Recoder Table Entry 12
Figure 6 Table Construction Data Flow 21
Figure 7 Energy Boundary Grouping Examples 23
Figure 8 Recoder Builder Module Level 1 Data Flow 28
Figure 9 Recoder Builder Module Global Function Structures 30
Figure 10 Recoder Builder Module Internal Function Structures 31

1 Introduction

1.1 Scope

This document describes the components that make up the Recoder Builder Module (RECODER). The Recoder Builder Module uses three Recoder Parameter Sets to generate Recoder Lookup Table Entries.

This document does not describe the overall design of the EDS, nor does it describe the operating details of each component in this module. It assumes that the reader is extremely familiar with the Flight Software Global Design, with the "C" programming language and with the architecture of the EDS.

This document should be of primary interest to Flight Software Engineering, Flight Hardware Engineering, Ground Software Engineering and Software Quality Assurance.

1.2 References

MIT 64-03001 Rev. G ..... "PCA-EDS Interface Control Document"
MIT 64-02006.01 Rev. D ..... "XTE EDS Hardware Functional Description and Requirements"
MIT 64-30102 Rev. E ..... "Flight Software Global Design"
MIT 64-30111.0801 Rev. C ..... "Typer Builder Module Specification" (TYPER)
MIT 64-30112.0601 Rev. D ..... "Science Thread Module Specification" (SCIENCE)
MIT 64-30110.2001 Rev. C ..... "Utility Module Specification" (UTIL)

1.3 Notations

This document assumes the following notations:

UINT8 ..... Refers to an 8-bit unsigned value.
UINT16 ..... Refers to a 16-bit unsigned value.

1.4 Module Perspective

The following diagram is a copy of the Level 1 data flow diagram from the **Flight Software Global Design**, Section 4.1.2. The Recoder Builder Module does not appear directly in this data flow, but has a scope which directly affects items indicated by the shaded areas:

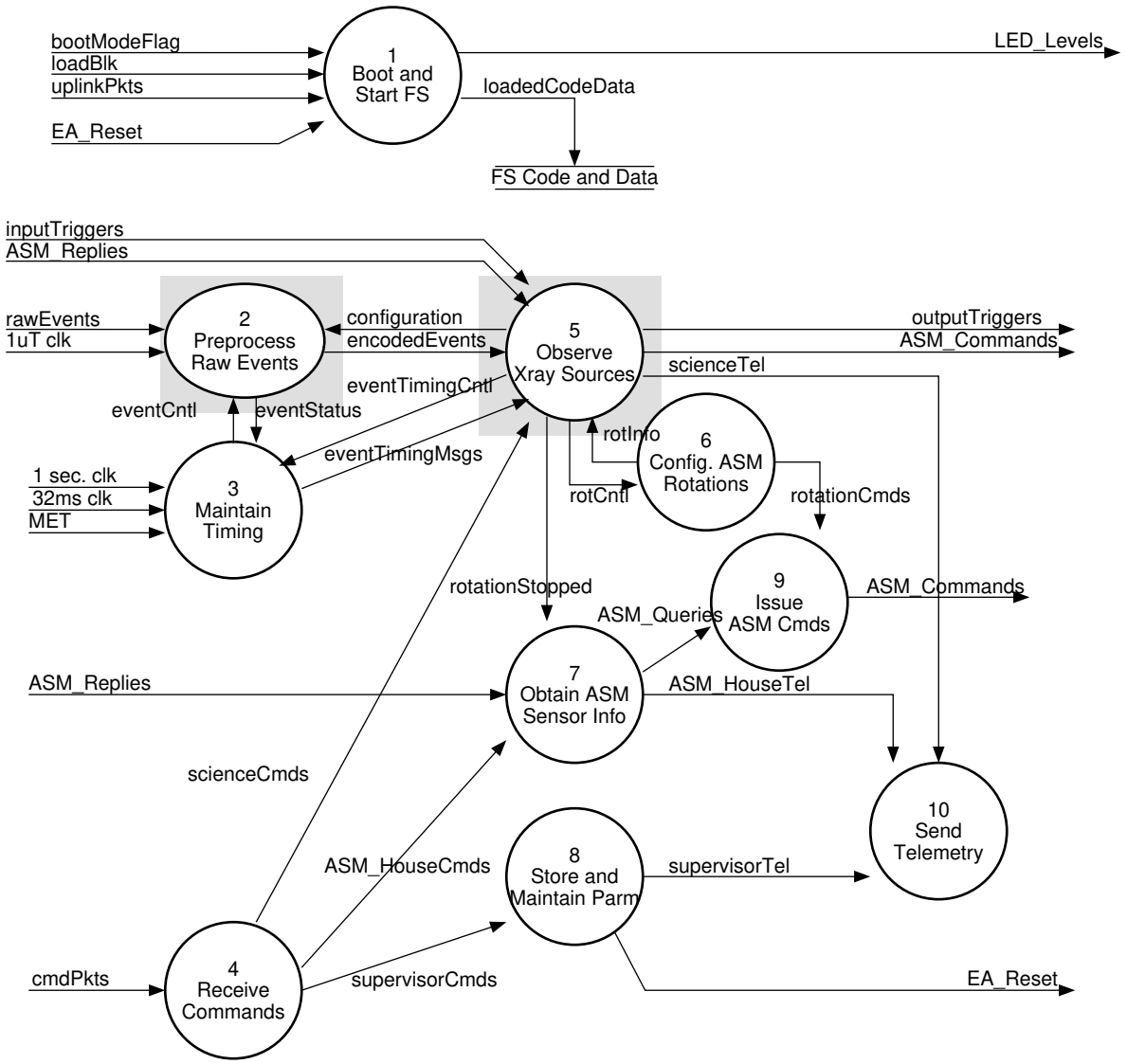

Figure 1 Module Perspective

1.5 Document Overview

Sections 1 and 2 provide an overview of the Recoder Builder Module, summarizing the purpose, services, context, and interfaces of the module. Section 3 describes all software structures and constants visible to modules outside of the Recder Builder Module. Section 4 describes, in detail, the Recoder Parameter Set formats. Section 5 describes the functions provided to other modules by the Recoder Builder Module. Section 6 describes some of the implementation details of the module. Section 7 describes which requirements are met by the module. Section 8 describes what types of tests need to be performed to ensure that the implementation of the module is operating properly. Appendix A lists items which are currently undefined, or that need to be reviewed.

1.6 Acronyms

Please refer to the **Flight Software Global Design Specification**, Section 1.5, for a list of acronyms and abbreviations.

2 Overview

2.1 Purpose

The purpose of the Recoder Builder Module is to generate Recoder Lookup Table Entries based on three parameter sets. The event processing hardware uses the Recoder Lookup Tables to adjust and classify the measured pulse height of an event. The Recoder Builder Module defines the format of the Parameter Sets, and provides a single function which uses these sets of parameters to build Recoder Lookup Table entries.

The Recoder Builder Module is designed to process events in a contiguous energy range, called "in-band" events. Events which have an energy outside of the given range are called "out-of-band" events.

2.2 Services

The Recoder Builder Module provides the following services:

1) Load hardware Recoder Table ..... The Recoder Builder Module is responsible for filling the contents of the Front End hardware Recoder Lookup Table, using ground specified pulse height correction, pulse height grouping, and output position parameters.

2) Provide Length of Parameters ..... The Recoder Builder Module is responsible for providing the length, in bytes, of each of the three parameter sets.

3) Check hardware Recoder Table ..... The Recoder Builder Module is responsible for checking the integrity of the Front End's hardware Recoder Lookup Table, using the original parameter sets for comparison.

2.3 Module Context

The following illustrates the Recoder Builder Module relationships with the rest of the Flight Software residing in an EA:

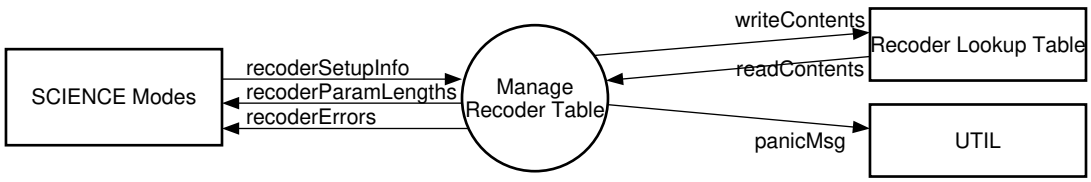

Figure 2 Recoder Builder Module Context Diagram

The Science Modes (see **Science Thread Module Specification**) use the Recoder Builder Module to load the contents of the Front End's hardware Recoder Lookup Table, using three sets of Recoder Parameters. If the Recoder Builder Module detects an error in the specified Recoder Parameter Sets, it issues a panic message (panicMsg) and resets the EA using the **Utility Module**. The Science Modes also use the Recoder Builder Module to obtain the length, in bytes, of the Recoder Parameter Sets (recoderParamLengths) and to test the integrity of the contents of the Front End hardware's Recoder Lookup Table (recoderErrors).

When building the Recoder Table, the Recoder Builder Module writes (writeContents) directly to the Front End's Recoder Lookup Table. When the Recoder Builder Module checks the integrity of the lookup table, it reads (readContents) the contents of the Front End's Recoder Lookup Table directly.

2.4 Theory of Operation

The following diagram illustrates the inputs and outputs of the pulse height recoding process:

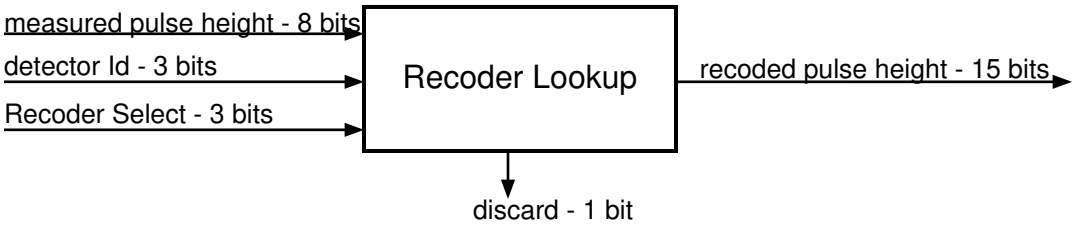

Figure 3 Pulse Height Recoding

The Recoder Lookup Table can be thought of as 8 "virtual" tables. As each event is processed, the Typer Lookup Table selects one of the eight Virtual Recoder

tables using the Recoder Select bits. The hardware then uses the selected Virtual Recoder Table to adjust and group the incoming event pulse height. The selected table may also tell the hardware to discard certain events. The hardware then combines the recoded pulse height with other information (such as time bin, event class, etc.) to form a recoded event word, which is used in later processing.

The Recoder Builder Module uses 3 Parameter Sets to build the entries stored in the Recoder Lookup Table. The Recoder Parameter Sets are generated by the ground and uplinked to an EA's memory. Before an observation starts, the Flight Software uses the Recoder Builder Module to build a Recoder Lookup Table based on these sets of parameters. The entries will be built directly into the Front End's recoder table. Recoder Parameter Sets are very closely tied to the input and output of the Recoder Lookup operation, and are described in detail in Section 4.

2.5 Data Structures and Constants

The Recoder Builder Module defines the structures of the Parameter Sets. It also uses the recoder table definition provided by the hardware.

2.6 Functions

The Recoder Builder Module defines the following functions:

RECODERbuild()	Generates Recoder Lookup Table Entries.
RECODERencoded_lengths()	Return Lengths of Recoder Parameters
RECODERcheck()	Check Integrity of Recoder Lookup Table

3 Data Structures and Constants

3.1 Hardware

3.1.1 Recoder Table (0x048000 flat address)

This is the lookup table used by the Front End PCA hardware to recode incoming pulse heights. The hardware discards any events which are so tagged in the table. The table contains 16,384 16-bit words (see **XTE EDS Hardware Functional Description and Requirements**, Section "FREGS and Observation Control : Lookup Table RAM" and "Figure 6 : PCA Event Processing").

3.1.2 PCA Recoder Input Event Word (14-bits)

The hardware uses this event word as an address into the Recoder Table. The following describes the table address bits generated by the hardware:

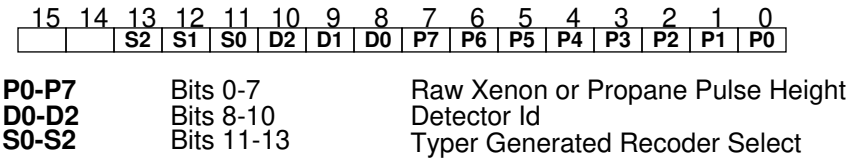

Figure 4 Recoder Table Input Event Word

Detector ID (3-bits) ..... This identifies the detector which produced the event. There are only 5 detectors in the system. Values 0 through 4 correspond to the detector Id. Values 5-7 indicate a hardware failure.

Recoder Select (3-bits) ..... This selects the Virtual Recoder Table to use. These bits are a subset of the Typer Lookup Table output word, selected by each event.

Pulse Height (8-bits) ..... This is the pulse height measured by the detector in either the main Xenon Layer or in the Propane Layer (see the **Typer Builder Module Specification** on how the Typer Lookup Table can distinguish between the two types of events).

The energy ranges for "in-band" and "out-of-band" events are indirectly specified for each detector/virtual table combination. The output for "in-band" events can also be detector/virtual table specific. The output for "out-of-band" events is the same for all detectors, but can vary in bit-position for each virtual table (see Section 3.2).

3.1.3 Recoder Table Entry (16-bits)

Values stored in the Recoder Table have the following format:

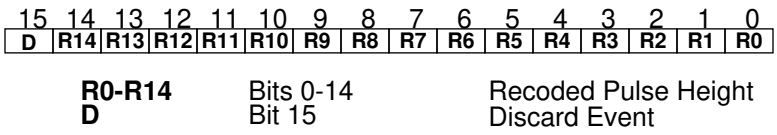

Figure 5 Recoder Table Entry

The "Discard Event" bit, when set, causes the hardware to discard the current event. The Recoder Builder Module can only discard "out-of-band" events (see Section 3.2).

3.1.4 Recoder Output Word (15 of 16 bits)

The hardware uses the lower 15 bits of the addressed Recoder Table Entry as the recoded pulse height of the event. The hardware logically OR's this word with the interval constant, data from the time tagging circuits, and with data from the event classification. The result is the event word that is sent to the DSP for further processing.

3.2 Recoder Parameter Set Structures

The Recoder Builder Module uses three Parameter Sets to build the final Recoder Lookup Table. The first Parameter Set specifies corrections to the detected pulse heights from each detector. Each set of corrections is specific to a given Virtual Recoder Table and Detector Id.

The second Parameter Set specifies how the corrected pulse heights are grouped. These groupings may vary for each Virtual Recoder Table. The correction and grouping parameters define the energy band location and width which selects "in-band" and "out-of-band" events.

The third Parameter Set specifies the value to use for "out-of-band" events, and the bit-position of the "out-of-band" code for each Virtual Table. The third parameter set also specifies the bit-position of the output group codes, Virtual Table Ids, and Detector Id's which are or'ed together to form the Recoder Output Word. These bit-positions are specific to each Virtual Recoder Table.

3.2.1 Pulse Height Correction Parameter Set (80 bytes)

The Pulse Height Correction Parameter Set consists of 8 sets of 5 shift and gain adjustment entries which may be used to calculate corrected pulse heights when combining results from different detectors. Each set corresponds to a single Virtual Recoder Table. Each entry within a set corresponds to one detector Id. The shift and gain adjustment values may take on values from -128 to 127. After corrections are applied, any events with a corrected energy less than 0 or greater than 255 are considered "out-of-band" events. Further selection of "out-of-band" and "in-band" events are controlled by the Pulse Height Grouping Parameter Set, described in Section 3.2.2. For a description of how the corrections are applied, refer to Section 4.2.

A single shift/gain adjustment entry is as follows:

[shift adjust] ..... Signed shift value adjustment (1-byte)
[gain adjust] ..... Signed gain adjustment (1-byte)

The Parameter Set consists of the following:

A given set of entries is indexed by detector id as follows:

[PCU 1 entry]..... Shift/Gain Adjustment Entry for Detector Id 0 (2-bytes)
[PCU 2 entry]..... Shift/Gain Adjustment Entry for Detector Id 1 (2-bytes)
[PCU 3 entry]..... Shift/Gain Adjustment Entry for Detector Id 2 (2-bytes)
[PCU 4 entry]..... Shift/Gain Adjustment Entry for Detector Id 3 (2-bytes)

[PCU 5 entry]..... Shift/Gain Adjustment Entry for Detector Id 4 (2-bytes)

The Pulse Height Correction Parameter Set consists of 8 correction parameter sets, each indexed by Virtual Table Id:

[Virtual Table 0 set]..... Corrections for Virtual Table 0 (10 bytes)
[Virtual Table 1 set]..... Corrections for Virtual Table 1 (10 bytes)
[Virtual Table 2 set]..... Corrections for Virtual Table 2 (10 bytes)
[Virtual Table 3 set]..... Corrections for Virtual Table 3 (10 bytes)
[Virtual Table 4 set]..... Corrections for Virtual Table 4 (10 bytes)
[Virtual Table 5 set]..... Corrections for Virtual Table 5 (10 bytes)
[Virtual Table 6 set]..... Corrections for Virtual Table 6 (10 bytes)
[Virtual Table 7 set]..... Corrections for Virtual Table 7 (10 bytes)

3.2.2 Pulse Height Grouping Parameter Set (from 2 to 274 bytes)

The Pulse Height Grouping Parameter Set consists of a variable number of fixed length entries. Each entry may be applied to one or more Virtual Recoder Tables. If a Virtual Table is never selected, no groupings are performed on that table, and each corrected pulse height produces a unique output code. Each entry has the following format:

[entry count] ..... Total Number of Entries (1-byte)
[pad] ..... Pad byte to maintain word alignment (1-byte)

Each entry has the following format:

[v7,v6,v5,v4,v3,v2,v1,v0] ..... Virtual Table Select Bit-field (1-byte)
Each bit corresponds to one of the Virtual Recoder Tables. "v0" corresponds to table 0, "v1" to table 1, and so on. If a bit is set, then the entry is applied to the corresponding Virtual Recoder Table. If no bits are set, then the entry is invalid.
[pad] ..... Pad byte to maintain word alignment (1-byte)
[boundaries ...] ..... Energy Boundary Bit Field (32-bytes)
Each bit corresponds to an energy level. Bit 0 of byte 0 corresponds to energy level 0, bit 1 of byte 0 to energy level 1, ... bit 7 of byte 0 to energy level 7, bit 0 of byte 1 to energy level 8, and so on. Each set bit corresponds to a boundary at the corresponding energy level. Events with an energy greater than or equal to the first boundary and less than the last boundary energy are considered "in-band" events, and are assigned group codes as follows: Events with an energy greater than or equal to the first boundary and less than the second are assigned an output value of zero, greater than or equal to the second but less than the third, 1, and so on. Events with an energy less than the first boundary or with an energy greater than or equal to the last boundary are "out-of-band" events. Events with an energy of 255 are always "out-of-band" events.

3.2.3 Output Position Parameter Set (18 bytes)

This Parameter Set consists of a 2-byte "out-of-band" code and 8 2-byte Position entries, one entry for each Virtual Recoder Table. Each Position entry consists of four position values; an Output Code bit-position, a Detector Id bit-position, a Virtual Table Id bit-position, and an "out-of-band" code bit-position. The bit-position values allows each Virtual Recoder Table to combine its Id, the Detector Id

and its output codes properly with the output of the Typer Table and Time-Bin Counter.

The following is the format of each bit-position entry:

[b,b,b,b,d,d,d,d,o,o,o,o,v,v,v,v] ..... Bit-field (2-bytes)
b ..... Bit position of "out-of-band" code (4-bits)
d ..... Bit position of Detector Id (4-bits)
o ..... Bit position of output codes (4-bits)
v ..... Bit position of Virtual Table Id (4-bits)

The following is the overall format of the Parameter Set:

["out-of-band" code] ..... Code to use for "out-of-band" events (2-bytes)
[Virtual Table 0 Bit Position Entries] ..... Bit Position Entries for Virtual Table 0 (2-bytes)
[Virtual Table 1 Bit Position Entries] ..... Bit Position Entries for Virtual Table 1 (2-bytes)
[Virtual Table 2 Bit Position Entries] ..... Bit Position Entries for Virtual Table 2 (2-bytes)
[Virtual Table 3 Bit Position Entries] ..... Bit Position Entries for Virtual Table 3 (2-bytes)
[Virtual Table 4 Bit Position Entries] ..... Bit Position Entries for Virtual Table 4 (2-bytes)
[Virtual Table 5 Bit Position Entries] ..... Bit Position Entries for Virtual Table 5 (2-bytes)
[Virtual Table 6 Bit Position Entries] ..... Bit Position Entries for Virtual Table 6 (2-bytes)
[Virtual Table 7 Bit Position Entries] ..... Bit Position Entries for Virtual Table 7 (2-bytes)

NOTE: To discard "out-of-band" events, use code RECODER_CONT_DISCARD_MASK (see Section 3.3.13) and an "out-of-band" code bit position of 0. The Detector Id and Virtual Table Id are not combined with "out-of-band" codes when processing "out-of-band" event entries.

3.3 Constants

3.3.1 RECODER_CHECK_DEFAULT (0x0ffff)

This defines the default value to use for the Recoder Table error count, offset and value (see RECODERcheck(), Section 5.3).

3.3.2 RECODER_NUM_RAW_ENERGIES (256)

This constant defines the maximum number of raw input pulse height levels sent by the PCA and used in the PCA Recoder Input Event Word. Since all corrected energies and group codes must be less than this value, this constant is also used to mark "out-of-band" events.

3.3.3 RECODER_NUM_VIRTUAL (8)

This constant defines the maximum number of Virtual Recoder Tables contained in the Typer Lookup Table.

3.3.4 RECODER_NUM_DETECTORS (5)

This constant defines the number of detectors in the PCA.

3.3.5 RECODER_NUM_DETECTOR_IDS (8)

This constant defines the number of possible detector Id's (For the actual number of physical detectors, see Section 3.3.4).

3.3.6 RECODER_ADDR_PULSE_HEIGHT_MASK (0x000ff)

This value, when "and"ed with a Recoder Table address, masks off all bits except the Pulse Height bits within the given address.

3.3.7 RECODER_ADDR_PULSE_HEIGHT_SHFT (0)

This value defines the number of bits by which to right shift a Recoder Table address to place the Pulse Height bits at bit position 0.

3.3.8 RECODER_ADDR_DETECTOR_ID_MASK (0x00700)

This value, when "and"ed with a Recoder Table address, masks off all bits except the Detector Id bits within the given address.

3.3.9 RECODER_ADDR_DETECTOR_ID_SHFT (8)

This value defines the number of bits by which to right shift a Recoder Table address to place the Detector Id bits at bit position 0.

3.3.10 RECODER_ADDR_TYPER_SELECT_MASK (0x03800)

This value, when "and"ed with a Recoder Table address, masks off all bits except the Typer Selection bits within the given address.

3.3.11 RECODER_ADDR_TYPER_SELECT_SHFT (11)

This value defines the number of bits by which to right shift a Recoder Table address to place the Typer Selection bits at bit position 0.

3.3.12 RECODER_CONT_OUTPUT_MASK (0x07fff)

This mask selects only Recoder Table output bits from a given Recoder Table Contents Word.

3.3.13 RECODER_CONT_DISCARD_MASK (0x08000)

This mask selects the "Discard" bit within a given Recoder Table Contents Word. This value can be used in the Output Position Parameter Set (see Section 3.2.3) to discard "out-of-band" events.

3.3.14 RECODER_POS_VIRT_MASK (0x0000f)

This mask selects the Virtual Table Id bit-position within an Output Position Parameter Set entry.

3.3.15 RECODER_POS_VIRT_SHFT (0)

This value defines the number of bits by which to right shift an Output Position Parameter Set entry to obtain the masked Virtual Table Id bit position value.

3.3.16 RECODER_POS_CODE_MASK (0x000f0)

This mask selects the Group Code bit-position within an Output Position Parameter Set entry.

3.3.17 RECODER_POS_CODE_SHFT (4)

This value defines the number of bits by which to right shift an Output Position Parameter Set entry to obtain the masked Group Code bit position value.

3.3.18 RECODER_POS_DET_MASK (0x00f00)

This mask selects the Detector Id bit-position within an Output Position Parameter Set entry.

3.3.19 RECODER_POS_DET_SHFT (8)

This value defines the number of bits by which to right shift an Output Position Parameter Set entry to obtain the masked Detector Id bit position value.

3.3.20 RECODER_POS_OUTBAND_MASK (0x0f000)

This mask selects the "out-of-band" code bit-position within an Output Position Parameter Set entry.

3.3.21 RECODER_POS_OUTBAND_SHFT (12)

This value defines the number of bits by which to right shift an Output Position Parameter Set entry to obtain the masked "out-of-band" code bit position value.

3.3.22 MSG_RECODER_DECODE_GROUPINGS_NULL

This message code is passed to UTILpanic() if the pointer to the Grouping Parameter Set passed to the internal function recoder_decode() is NULL. The message should read something like:

"PANIC: ---recoder_decode() -- Grouping Parameter Set pointer is NULL"

3.3.23 MSG_RECODER_DECODE_POSITION_NULL

This message code is passed to UTILpanic() if the pointer to the Output Position Parameter Set passed to the internal function recoder_decode() is NULL. The message should read something like:

"PANIC: ---recoder_decode() -- Output Position Parameter Set pointer is NULL"

4 Recoder Table Construction

4.1 Construction Algorithm

There are three main goals of the Recoder Lookup Table. The first is to correct for pulse height gain and shift changes in the detectors when combining pulse heights from the different detectors, given either a Xenon Pulse Height or Propane Pulse Height (selected via the Typer Table using the Virtual Table Id). The second use is to recode the corrected pulse heights to reduce the amount of information produced by the pulse height measurements. The third is to format the resulting information so that it can be combined with information produced by the Typer Table and Time Bin counter and to either discard "out-of-band" events or to combine these events into a single group.

The Recoder Table Entry construction process uses three Parameter Sets to describe the contents of the Recoder Lookup Table. First, the process uses the Pulse Height Correction Parameter Set to generate an energy mapping which compensates for gain and energy shift differences between the detectors, given that the event was detected in either the Xenon Layer or the Propane Layer. Second, it uses the Pulse Height Grouping Parameters to group the corrected pulse heights into an ascending set of output codes. Third, it uses the Output Position Parameters to position the resulting corrected and mapped energies into the appropriate output

bits and combine the results with a positioned Virtual Table Id. The following diagram attempts to illustrate the process of filling the Recoder Lookup Table:

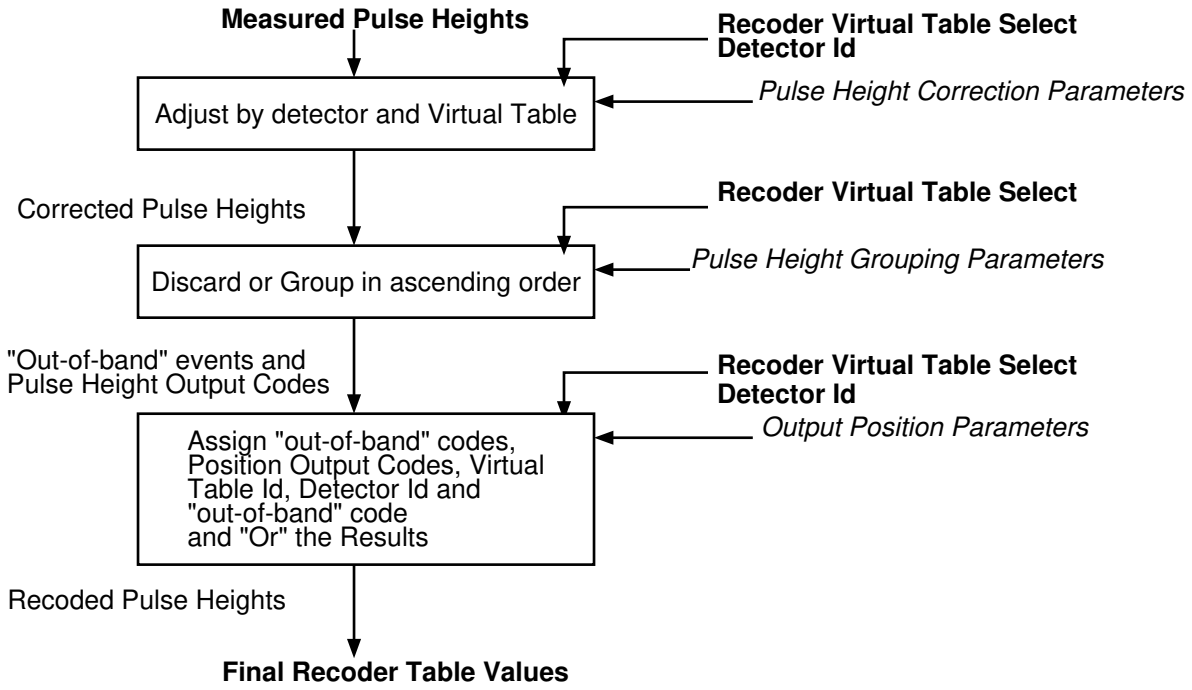

Figure 6 Table Construction Data Flow

4.2 Pulse Height Correction

All detectors are assumed to have a linear response in both the Xenon and Propane Layers. Differences in gain and offset may occur between the detectors and layers. The Pulse Height Correction Parameter Set specifies two adjustment values for each detector and virtual table id. Given an initial one-to-one table mapping from a given detector, the first adjustment value specifies how much to add to or subtract from all of the table's outputs for that detector. The second value modifies the slope of the initial 1:1 response, in plus or minus 1/256 increments (NOTE: The shift adjustment is applied after the slope adjustment).

If a corrected energy level is less than zero or greater than 255, then the corresponding event is considered "out-of-band" (NOTE: The Pulse Height Groupings consider all events with energy equal to 255 as "out-of-band"). Values are rounded to the nearest integer value. Values exactly between two integer values are rounded up. An identity table is achieved by setting both adjustment values to zero.

4.3 Pulse Height Grouping

The Recoder Builder Module uses the Pulse Height Grouping Parameter Set to group contiguous corrected pulse height energies and assign them ascending output codes. It also uses this Parameter Set to mark events with corrected energies below and above specified limits as "out-of-band" events.

The Pulse Height Grouping Parameter Set consists of a variable number of fixed length entries. Each entry consists of a 32 byte (256-bit) energy boundary mask and a Virtual Table Selection Mask. The energy boundary mask determines how the module should group the corrected input energies. Each bit in the mask corresponds to one of the 256 possible energy levels. If a bit is set, then the corresponding energy level is a boundary. Energies between two boundaries, or at the lower energy boundary are grouped and assigned the same output code. Events with energies less than the lowest energy boundary are marked as "out-of-band". Events with energies at or greater than the highest energy boundary are also marked as "out-of'band". Ascending output codes are assigned to the groups, starting with zero for the group at the lowest energy level. All events with an energy at 255 are marked as "out-of-band". The Virtual Table Selection Mask controls which virtual tables are affected by the entry.

If a given Virtual Table has no corresponding grouping entry, the corrected pulse height values for that Virtual Table are not modified by the Pulse Height Grouping Parameter Set (i.e. If no corrections are applied, the result is a pulse-height identity matrix. If corrections are applied, the result is a corrected pulse-height identity matrix). In this case, events with an energy at 255 may or may not be "in-band", depending on the pulse-height corrections (see Section 4.2).

The following diagram attempts to illustrate the behavior of these "energy boundaries." The horizontal axes represent the corrected pulse heights, the vertical axes represent the generated output code. The numbers at the bottom of each diagram represent the energy mask bits, and the shaded areas represent corrected pulse heights which cause the event to be discarded. The diagram illustrates only

16 input energies, although, in reality, there are 256 possible input energies. The behavior of energy 255 is represented by 15.

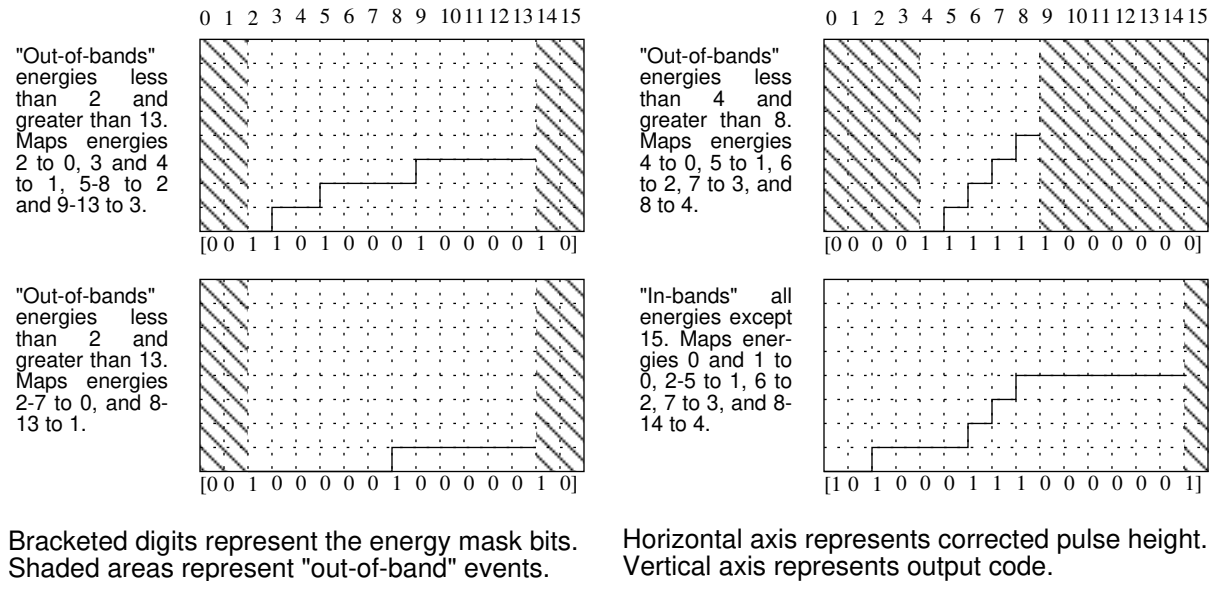

Figure 7 Energy Boundary Grouping Examples

4.4 Output Position

Each Virtual Table has a corresponding entry in the Output Position Parameter Set, which is described in Section 3.2.3. The Parameter Set specifies a single "out-of-band" code followed by 8 bit-position entries, where each entry contains four bit-position values. The "out-of-band" code applies to all Recoder Table entries which are marked as "out-of-band" by the Pulse Height Correction Parameters or Grouping Parameters. The first bit-position value specifies the bit-position of the least significant bit of the Virtual Table Id. If this value is 15, then the Virtual Table Id does not appear in the output. The second bit-position value specifies the bit-position of the least significant bit of the pulse height output codes (not including the "out-of-band" code). If this value is 15, then all of the output codes are set to zero. If this value is zero, then the output codes are not affected. The third value specifies the bit-position of the least significant bit of the Detector Id. If this value is 15, then the Detector Id does not appear in the output. The fourth value specifies the bit-position of the least-significant bit of "out-of-band" code.

To discard "out-of-band" events, the "out-of-band" code must be set to RECODER_CONT_DISCARD_MASK (see Section 3.3.13) and the "out-of-band" bit position must be set to 0. NOTE: It is possible to inadvertently (or intentionally) discard "out-of-band" events by specifying a non-zero "out-of-band" code and by shifting a 1 into bit 15 (the "discard" bit of the Contents Word). This condition will not occur with Detector Id's, Virtual Table Id's or Group Codes, since the algorithm masks off bit 15 on "in-band" events.

The following attempts to illustrate the application of the Output Position Parameters. The following table shows an example set of Output Position Parameters:

Out-of-Band Code	5			
	Out-of-Band Position	**Detector Id Position**	**Group Code Position**	**Virtual Table Id Position**
Virtual Table 0	15	15	15	15
Virtual Table 1	15	8	0	11
Virtual Table 2	0	15	15	15
Virtual Table 3	15	8	0	15
Virtual Table 4	0	1	4	12
Virtual Table 5	13	6	2	11
Virtual Table 6	15	8	0	11
Virtual Table 7	15	8	0	11

The following diagram illustrates the accepted event address fields, followed by the Out-of-Band code used for each Virtual Table:

	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
In-Band 0	0															
Out-of-Band 0	1	0														
In-Band 1, 6, 7	0		Virtual Table Id (i.e. 1, 6 or 7)			Detector Id			Energy Group Code							
Out-of-Band 1, 6, 7	1	0														
In-Band 2	0															
Out-of-Band 2	0													1	0	1
In-Band 3	0					Detector Id			Energy Group Code							
Out-of-Band 3	1	0														
In-Band 4	0	Virtual Table Id = 4			Energy Group Code								Detector Id			0
Out-of-Band 4	0													1	0	1
In-Band 5	0		Virtual Table Id = 5			0		Detector Id			Energy Group Code				0	
Out-of-Band 5	1	0	1	0												

Note that the number of Energy Group Codes can be up to 256 for all Virtual Tables except 5, which requires a maximum of 16 Energy Group Codes. Also note that, in Virtual Table 0, 1, 3, 5, 6 and 7, Out-of-Band events will be rejected (the Out-of-Band code was shifted such that the most significant 1 is in the Discard Event slot (bit 15).

5 Functions

5.1 RECODERbuild () - Build Recoder Table Entries

Usage

void RECODERbuild (UINT8 **corrections*, UINT8 **groupings*, UINT8 **positions*);

Description

RECODERbuild() uses three Parameter Sets to construct the Recoder Table Entries. "*corrections*" points to a Pulse Height Correction Parameter Set, "*groupings*" points to a Pulse Height Grouping Parameter Set, and "*positions*" points to an Output Position Parameter Set. The final table is written directly into the Front End hardware's Recoder Lookup Table. If "*corrections*" is NULL, no pulse height corrections will be applied to the Recoder Table.

If an error is encountered while decoding the parameters, RECODERbuild() will issue a panic message and reset the EA, using the **Utility Module** function UTILpanic(). The following lists the possible error messages:

MSG_RECODER_DECODE_GROUPINGS_NULL ..... "*groupings*" is NULL.
MSG_RECODER_DECODE_POSITION_NULL ..... "*positions*" is NULL.

Return Values

NONE - FUNCTION IS VOID.

Caveats

1) "*corrections*" must be NULL or must point to data which conforms to the structure described in Section 3.2.1. "*groupings*" must point to data which conforms to the structure described in Section 3.2.2. "*positions*" must point to data which conforms to the structure described in Section 3.2.3.

2) The DSP MUST be held in a reset state while calling this function.

See Also

Utility Module Specification

5.2 RECODERencoded_lengths() - Lengths of Parameter Sets

Usage

```
void RECODERencoded_lengths (
    UINT8 *corrections,    UINT16 *corrections_length,
    UINT8 *groupings,  UINT16 *groupings_length,
    UINT8 *positions, UINT16 *position_length);
```

Description

RECODERencoded_lengths() supplies the lengths, in bytes, of each of the specified Recoder Parameter Sets. "*corrections*" points to a Pulse Height Correction Parameter Set, "*groupings*" points to a Pulse Height Grouping Parameter Set, and "*positions*" points to an Output Position Parameter Set. "*corrections_length"*, "*groupings_length*" and "*position_length*" each point to an unsigned 16-bit variable. RECODERencoded_lengths() stores the length of the corresponding Parameter Set into each of these variables. If any pointer to a Parameter Set is NULL, RECODERencoded_lengths() stores a value of 0 in the corresponding variable.

If "*corrections_length"*, "*groupings_length*" or "*position_length*" are NULL, no value will be written to the corresponding variable.

Return Values

NONE - FUNCTION IS VOID.

Caveats

1) "*corrections*" must be NULL or must point to data which conforms to the structure described in Section 3.2.1. "*groupings*" must point to data which conforms to the structure described in Section 3.2.2. "*positions*" must point to data which conforms to the structure described in Section 3.2.3.

See Also

NONE

5.3 RECODERcheck() - Check Integrity of Recoder Table

Usage

UINT16 RECODERcheck (UINT8 *_corrections_, UINT8 *_groupings_, UINT8 *_positions_, UINT16 *_err_offset_, UINT16 *_err_val_);

Description

RECODERcheck() decodes the passed Parameter Sets, pointed to by "*corrections*", "*groupings*", and "*positions*" into the Science Buffer, and performs a word-by-word comparison of the newly decoded table with the Front End hardware's Recoder Lookup Table. If a discrepancy is found, RECODERcheck() fills the contents of *"err_offset*" with the word index in the hardware Recoder Table that had the error, and fills the contents of "*err_val*" with the actual value discovered at that location. It then increments an error counter. If more errors are found, RECODERcheck() again increments the counter, without reporting the new offset and value which disagreed with the Recoder Parameters. After the entire table has been checked, RECODERcheck() returns the value of the error counter. If no discrepencies are found, RECODERcheck() stores RECODER_CHECK_DEFAULT into the locations pointed to by "*err_offset*" and "*err_val*" and returns an error count of 0. "*corrections*" is either NULL, or points to a Pulse Height Correction Parameter Set, "*groupings*" points to a Pulse Height Grouping Parameter Set, and "*positions*" points to an Output Position Parameter Set. If an error is encountered while decoding the parameters, RECODERcheck() will issue a panic message and reset the EA, using the **Utility Module** function UTILpanic(). The following lists the possible error messages:

MSG_RECODER_DECODE_GROUPINGS_NULL ..... "*groupings*" is NULL.
MSG_RECODER_DECODE_POSITION_NULL ..... "*positions*" is NULL.

Return Values

The total number of discrepancies between the Recoder Parameter Sets and the Front End hardware's Recoder Lookup Table.

Caveats

1) RECODERbuild() must be called before calling this function.

2) The DSP MUST be held in a reset state while calling this function.

3) The contents of the Science Buffer will be corrupted by calling this function.

See Also

Utility Module Specification

6 Detailed Design Considerations

6.1 Data Flow

The following diagram illustrates the level 1 data flow within the Recoder Builder Module:

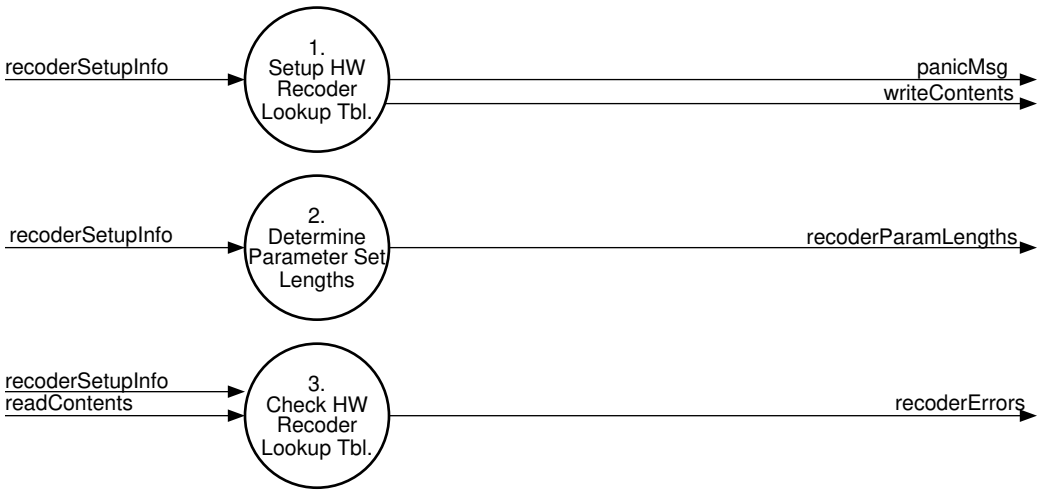

Figure 8 Recoder Builder Module Level 1 Data Flow

Before the start of an observation, the Recoder Builder Module is called to decode (bubble 1) a set of Recoder Parameters (recoderSetupInfo) into the Front End Hardware's Recoder Lookup Table (writeContents), and to determine (bubble 2) the lengths (recoderParamLengths), in bytes, of each of the Recoder Parameter Sets (recoderSetupInfo). If an error is encountered while building the Front End's Recoder Table, the Recoder Builder Module issues a panic message and resets the EA (panicMsg).

After the observation has terminated, the Recoder Builder Module is called to test the integrity of the Front End's Recoder Lookup Table (bubble 3). The Recoder Builder Module decodes the supplied Recoder Parameter Sets (recoderSetupInfo) into the Science Data Buffer (not shown) and then compares the buffer with the contents of the Front End's Recoder Table (readContents). The Recoder Builder Module then reports any differences (recoderErrors) to the caller.

6.2 Control Considerations

6.2.1 Pulse Height Correction Algorithm

This section describes how energy correction mappings are produced using the Recoder Correction Parameter Set. The Pulse Height Correction Parameter Set specifies a "gain" and "shift" adjustment to apply to the measured raw Xenon or Propane Pulse Heights as produced by each detector of the PCA (NOTE: The Typer Lookup Table is responsible for selecting different Virtual Recoder Tables based on whether the event was detected in the Xenon Layer or in the Propane Layer). Initially, the mapping is 1 to 1 (i.e. has a slope of 256/256). The "gain" adjustment specifies how much to add or subtract to the response's slope, in 1/256 increments. For example, if "gain" is set to -128, the slope of the corrected energies is (256 - 128)/256, or 0.5. If "gain" is set to +127, then the slope is (256 + 127)/256, or about 1.496. The "shift" parameter specifies how much to add or subtract from the "gain" corrected energies.

To produce a pulse height correction map, compute the "slope" of the map by adding the "gain" parameter to 256. Then iteratively compute a "gain" corrected energy. Start with a "gain" corrected energy of zero and a raw input energy of zero. Using integer arithmetic, compute the corrected energy by adding 128 to the "gain" corrected energy, and by dividing the result by 256 and adding "shift." Adding 128 to the "gain" corrected energy before the division causes the result of the division to be rounded up to the nearest integer value (i.e. 1.4 -> 1, whereas 1.5 -> 2). After the corrected energy has been computed and stored, compute the next "gain" corrected energy by adding "slope" to the previous "gain" corrected energy value, and repeat for each of the raw input energies.

6.2.2 Pulse Height Grouping Algorithm

This section describes how energies are grouped based on the Recoder Grouping Parameter Set. The Pulse Height Grouping Parameter Set is an array of entries, where each entry consists of an 8-bit Virtual Table Select bit-field and a 256-bit (32 byte) Energy Boundary bit-field. The Virtual Table Select bit-field selects one or more Virtual Recoder Tables, and the Energy Boundary bit-field specifies which energies are grouped together with the same output energy code. Each bit in the the Energy Boundary bit-field corresponds to a given input energy (i.e. bit 45 corresponds to input energy 45). The Energy Boundary bit-field is used to produce an input energy to a group code mapping array, where the contents of the array specify the output group code for each input energy.

The mapping array is produced by scanning the Boundary bit-field, starting from bit 0, while maintaining an incrementing group code. Each time a bit in the Boundary bit-field is encountered which is set to "1", the output group code is incremented. As each bit in the Boundary bit-field is scanned, the current group code is stored in the mapping array entry corresponding to the energy which corre-

sponds to the bit in the Boundary bit-field. The entries in the mapping array corresponding to bits before the first "1" was encountered in the bit-field and at and after the last "1" in the bit-field are marked as "out-of-band" events.

6.3 Module Structure

6.3.1 Global Function Structures

This section describes the structure of the Recoder Builder Module's globally accessible functions. Bolded boxes and text refer to function provided by modules other than the Recoder Builder Module:

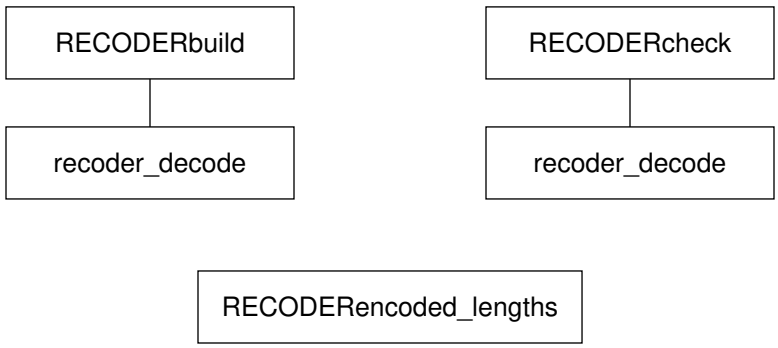

Figure 9 Recoder Builder Module Global Function Structures

The following lists the functions shown in Figure 9. The details of each function's operation and arguments are described in the Recoder Builder Module's detailed design document.

RECODERbuild ..... Fills Front End Recoder Lookup Table.
RECODERencoded_lengths ..... Get length, in bytes, of each Parameter Set.
RECODERcheck ..... Check integrity of Front End Recoder Lookup Table

6.3.2 Internal Function Structures

This section describes the structure of the Recoder Builder Module's internal functions. Bolded boxes and text refer to function provided by modules other than the Recoder Builder Module:

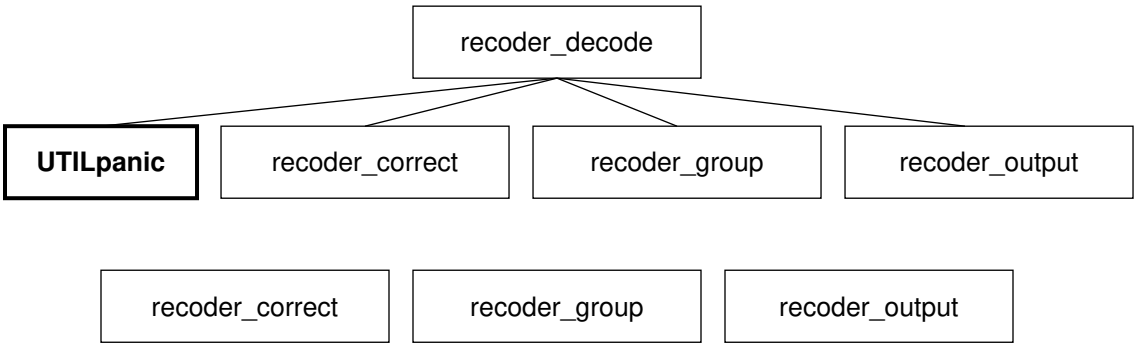

Figure 10 Recoder Builder Module Internal Function Structures

recoder_decode ..... Decode Recoder Parameter Sets into specified table
recoder_correct ..... Generate a Corrected Energy lookup table based on the specified Corrections Parameters, Detector Id and Virtual Table Id.
recoder_group ..... Generate an Energy Group lookup table based on the specified Grouping Parameters and Virtual Table Id.
recoder_output ..... Form a Recoder Table Contents Word based on the passed Positions Parameters, Detector Id, Virtual Table Id and Energy Group Code.

6.4 Optimization Considerations

If the worst case execution time for recoder_decode is greater than 1 second, recoder_decode() must be optimized or modified to prevent the Science Thread from executing for longer than 1 second without relinquishing control (inserting a Twait(0L) call for each Detector Id or Virtual Table Id may be sufficient).

7 Requirements Traceability

7.1 Science Requirements

The Recoder Builder Module addresses all PCA Science Requirements associated with the ability of the ground to control how events are grouped based on their measured Xenon or Propane Pulse Height (pulse height recoding). The requirements which have been addressed by this document are listed in the **Requirements Traceablity Matrix**.

This module does not address the PCA Science Requirements associated with Event Type Classification. These are handled in the **Typer Builder Module Specification**.

8 Testing Considerations

8.1 Unit Testing

To perform unit tests on this module, the following equipment is required:

a. An Event Analyzer equipped with the Intermetrics ROM Monitor Debug Board

b. A workstation running the Intermetrics debugger

c. An RS-232 serial cable between the ROM Monitor's serial port and the workstation

To unit test this module, the tester must link and locate this module's functions to some EA RAM memory location. Using a resident monitor/debugger, the tester should first verify the operation of all functions internal to this module, and then verify the interactions between all functions internal to the module. These tests should verify all branches, calls, and legal and illegal parameters. The tester should then verify all calls made to functions outside of this module, and ensure that the passed arguments are always within the limits specified by the corresponding module specifications. The tester should also verify that the caller always handles values returned by these functions in an appropriate manner.

In addition to these tests, the tester should perform the following:

a. Test Pulse Height Correction by detector and Virtual Table. Generate a set of Correction Parameter Sets and verify that the energy correction mappings are correct. The sets of parameters must cover situations where the "gain" and "shift" parameters are set to zero, and their minimum and maximum values. Other values must also be tested. Generate Correction Parameter Sets using different parameters for the different detectors and Virtual Table Selections and ensure that the correction parameters are applied independently for each detector and Virtual Table.

b. Test Pulse Height Grouping by Virtual Table. Generate a set of Grouping Parameter Sets and verify that the energy group mappings are correct. The sets of parameters must test situations where no energy groups are produced (all events are "out-of-band"), one energy group is produced, and where the maximum number of energy groups are produced. Other group configurations must also be tested. Generate Grouping Parameter Sets using different parameters

for each Virtual Table and ensure that the grouping parameters can be applied independently for each Virtual Table.

c. Test Output Code formation. Generate a set of Output Position Parameter Sets and verify that the Recoder Table Contents are properly formed. The sets of parameters must verify situations where the detector Id is and is not output, where the virtual table Id is and is not output, where the output energy code is and is not output. The tests should also verify conditions where no "out-of-band" events are generated, where "out-of-band" events are discarded, and where "out-of-band" codes in different Virtual Table's are output in different bit positions. The tests should also verify that the detector Id, virtual table Id and output energy code are positioned in a variety of bit locations within the recoded pulse height word. These locations should include illegally large shifts to ensure that the discard bit is succesfully masked.

d. Test combinations of Pulse Height correction, grouping and output code positioning. Ensure that the formed Recoder Lookup Table is properly formed based on the specified parameter sets. Generate a set of Parameter Sets which "pass-through" the detector Id, virtual table Id, and input energy.

e. Generate Parameter Sets which apply different corrections to all detector/virtual id combinations, produce the maximum number of group codes, use a separate Grouping Parameter Entry for each virtual table, and which supply the detector Id, virtual table Id and energy group code in the final Recoder Contents Word. Measure the execution time of recoder_decode() and ensure that it takes no longer than 1 second to produce the final Recoder Lookup Table. If the time taken is greater than 1 second, either optimize the decoding functions, or modify the design to yield control while decoding the table. Also ensure that recoder_decode() properly fills the appropriate locations with the proper values in the Recoder Table.

8.2 Integration Testing

For integration tests, the following equipment is required:

a. An Event Analyzer

b. A populated System Manager board

c. A logic analyzer

d. Hardware which can:

   1. Drive the MinorTick and MajorTick clock signals (S/C Simulator)

   2. Receive telemetry packets from the EA (S/C Simulator).

3. Block telemetry packets from the EA (i.e. stop receiving packets).

4. Send simulated PCA events to the EA (PCA-Simulator).

The integration tests should verify the operation of the Recoder Builder Module in conjunction with the Science Thread Module and with the event processing hardware. These tests should include:

a. Verify that the Recoder Builder Module properly builds Recoder Tables under the control of the Science Thread.

b. Verify that the Recoder Builder Module properly detects single-bit corruptions of the built Recoder Table.

c. Verify that the Recoder Builder Module properly detects outright corruption of the built Recoder Table.

d. Verify that the Science Buffer is not in use while the Recoder Builder Module checks the Recoder Table, and that the DSP is held in a reset state.

e. Verify the operation of the Front End Recoder Table using fabricated PCA event words and all Science Modes.

f. Ensure that the Recoder Table properly discards undesired (i.e. "out-of-band") events.

g. Ensure that the Recoder Table properly handles "out-of-band" events.

h. Ensure that the appropriate Virtual Recoder Table is properly selected by the Typer Table (this may be part of the **Typer Builder Module** tests).

i. Test the Recoder Table using the PCA Standard Data Modes, as defined in the **PCA-EDS Science Requirements Specification**.

j. Exercise the Recoder Table using all PCA Science Modes and using a variety of configurations. Ensure that the Recoder Table can be configured successfully, such that the pulse height recoding requirements for all modes, defined in the **PCA-EDS Science Requirements Specification**, are met.

Appendix A

A.1 Items Under Review

This section lists items and issues which need to be resolved:

Document Reference	Section	Resolution Expected by
NONE		